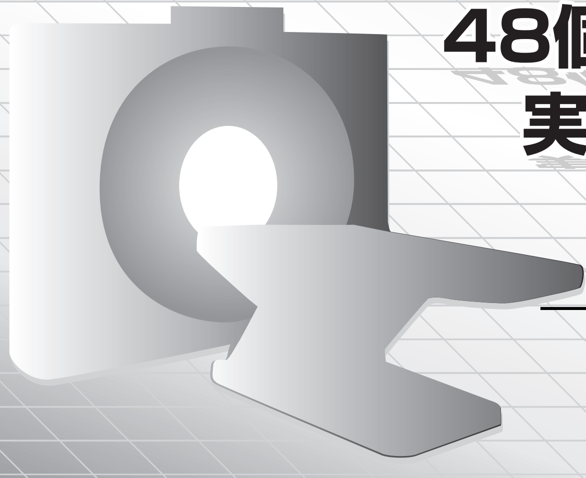


48個のALUが並列動作する実効性能10.2GFLOPSの画像処理プロセッサを開発

――医療機器市場の要求とプロセッサ開発の経緯

デバイスの記事　システムの記事

Y

ここでは，医療機器向け画像処理プロセッサの開発事例を紹介する．筆者らは，実効性能が10.2GFLOPS（1GFLOPSは，1秒当たり10億回の浮動小数点演算）の画像処理プロセッサを開発した．医療機器における画像処理では，再構成演算やボリューム・レンダリングなど，非常に負荷の重い処理を行うケースが多い．そのため，筆者らは高性能なカスタム・プロセッサを開発している．今回のプロセッサ開発では，多並列化やベクトル型などの手法を駆使して，性能の向上を達成した．

（編集部）

現代のプロセッサ技術は大きな転換期を迎えています．半導体プロセスの微細化によって性能を向上できる，という期待は小さくなってきました．また，これまではパイプラインの段数を増やすことが性能向上の常とう手段でしたが，すでに48段パイプラインのプロセッサが登場しており，さらなる性能の向上をねらうには根本的なアーキテクチャの変更が必要になります．

ここではこれからのプロセッサ技術について考察し，その一つの方向性の例として，筆者ら（米国TeraRecon社[注]）が医療機器向けの画像処理プロセッサとして開発した「XTrillion3.0」の技術と開発の経緯を説明します．

注：TeraRecon社は1997年5月にシリコンバレー（米国カリフォルニア州San Mateo）にて，日本人医師により設立された医療系ベンチャ．従業員数は米国，日本を中心に約230名．主力の3次元医療画像診断装置「アクエリアス・シリーズ」は世界で1,000台を超える稼働実績を持つ．信号処理・画像処理関連の技術を保有しており，精密な医療画像の提供を通じて医療の質を向上させていくことを目ざしている．同社の日本支店では，主としてプロセッサを含むハードウェアの開発を行っている．

● プロセッサにはさまざまな分類法がある

プロセッサにはいろいろな種類があります．分類方法を**表1**に示します（このとおりに，きちんと分かれるわけではないが…）．

現代のプロセッサのほとんどはパイプライン化されています．例えば，たんにALU（arithmetic logical unit）がパイプライン化されているだけのものは，ベクトル・プロセッサとは呼びません．チェイニング（ALUの演算結果をそのまま次のALUに渡す機能）ができるか，ベクトル・レジスタがあるかなどで，スカラ・プロセッサかベクトル・プロセッサかを分類します．

ベクトル・プロセッサの場合，ベクトル演算に対応するためにRAMやレジスタ・ファイルが通常のものよりも高機能になっています．ベクトル型スーパ・コンピュータに分類されるNECの「SXシリーズ」や地球シミュレータなどは，内部に非常に高度なスケジューラを実装しています．そのため，これらのプロセッサはベクトル型ではありますが，スカラ演算も高機能です．

では，パソコンやゲーム機などで使われている主流のマイクロプロセッサはどのように進化しているのでしょう．例えばPowerPCプロセッサは，内部にAltivecと呼ぶ機構を備えています．この中にはベクトル・レジスタが含まれており，ベクトル・プロセッサとしての側面を持っています．x86は高度なOS環境と組み合わせて使うことを前提としており，スカラ演算の強化を主軸において改良が続けられています．同じCPUコアを二つ実装する「デュアル・コア」は，その現れであると筆者らは考えています．

KeyWord　医療機器，画像処理プロセッサ，メモリ・バンド幅，キャッシュ・リフィル，ベクトル・プロセッサ，MIMD，FLOPS，ボリューム・レンダリング，XTrillion，Amplify ASIC

表1
プロセッサの分類

分　類	CISC(complex instruction set computer)	RISC(reduced instruction set computer)	LIW(long instruction word)
特　徴	● 命令長が可変 ● 複雑な命令がある ● 命令メモリが少なくて済む ● 速度が上げにくい	● 命令長が固定 ● シンプルな命令のみ ● 命令メモリが多い	● 命令に複数の演算命令が含まれている ● 命令のビット・フィールドが長いものは，とくにVLIW(very long instruction word)と呼ぶ(64ビットより長いのが一般的) ● スケジューリングはコンパイラが担当．ほかはハード・ワイヤード論理にて処理

(**a**) 命令の表現方法による分類

分　類	スカラ・プロセッサ(スーパスカラ・プロセッサ)	ベクトル・プロセッサ
特　徴	● 分岐が頻繁に発生する演算 ● 演算長が不確定	● 分岐がほとんどない(実際は演算終了までありえない) ● 演算長が確定

(**b**) 得意とする演算による分類

分　類	SIMD(single instruction stream-multiple data stream)	MIMD (multiple instruction stream-multiple data stream)
特　徴	● 一つの命令で全ALUが同じ演算を実行する	● 全ALUが独立に演算でき，それぞれ命令を持っている

(**c**) 並列演算方法による分類

表2
性能とバンド幅

プロセッサ	ピーク性能	FSBのバンド幅	メモリ・バンド幅	ピーク性能の12倍とFSBバンド幅注の比
Pentium 4 EXTREME EDITON(3.73GHz動作)	14.92GFLOPS	8.52Gバイト/s	10.6Gバイト/s	8.52Gバイト/s / (14.92GFLOPS × 12) = 4.8 %
Opteron 154(2.8GHz動作)	11.20GFLOPS	―	5.38Gバイト/s	5.38Gバイト/s / (11.20GFLOPS × 12) = 4.0 %
Xenon(3.2GHz動作)	115GFLOPS	21.6Gバイト/s	22.4Gバイト/s	21.6Gバイト/s / (115GFLOPS × 12) = 1.6 %
Cell(3.2GHz動作)	218GFLOPS	76.8Gバイト/s	25.6Gバイト/s	25.6Gバイト/s / (218GFLOPS × 12) = 1.0 %

注：OpteronとCellはメモリ・バンド幅で計算

プロセッサは，並列演算の方式によっても分類できます．Pentiumプロセッサのマルチメディア拡張命令であるSSE(Streaming SIMD Extensions)などはSIMDに分類されます．最新のGPU(graphics processing unit)は，一部にMIMDを採用しています．

● 演算性能を上げる方法はひと通りではない

表1の分類を踏まえて，プロセッサの演算性能を引き上げるためにどのような方法が残されているのかを考察してみます．

1) 多並列化

多並列化の代表格にはXenon(米国IBM社と米国Microsoft社の共同開発)，Cell(IBM社，ソニー・コンピュータエンタテインメント，東芝の共同開発)，Niagara(米国Sun Microsystems社)，Pentium D(米国Intel社)，Dual Core Opteron(米国Advanced Micro Devices社)などがあります．XenonやCellは家庭用ゲーム機で使われるものなので，名まえをご存じのかたも多いかと思います．しかし，今後，さらなる多並列化が可能かどうかはLSIの集積度向上，すなわち半導体プロセスの進歩に依存しています．

2) キャッシュ・メモリの大容量化

実際のアプリケーションにおいて，プロセッサの実効性能は，最適化なしには目標を達成できないことが多々あります．これは頻繁なキャッシュ・リフィル(キャッシュ・メモリの中身を入れ替える処理)が発生し，大きなオーバヘッドが生じるからです．この頻度を下げるため，キャッシュ・メモリ・サイズを大きくすることが求められています．現在，16Mバイトのキャッシュ・メモリを搭載しているプロセッサがありますが，現実にはまだまだ足りません．この方法も，半導体の集積度に左右されます．

3) メイン・メモリとプロセッサの間のバンド幅の拡大

表2に，ピーク性能，FSB(front side bus)バンド幅，メモリ・バンド幅の例を列挙します．Pentium 4とOpteronはシングル・コアのプロセッサ，XenonとCellはマルチコアのプロセッサです．

単精度浮動小数点で1FLOPSを達成するには，1項＝4バイト/sとすると，2項演算では読み出しに8バイト，結果の書き込みに4バイト必要です．つまり，ピーク性能をできるだけ現実の数字に近づけたければ，メモリ・バンド幅がピーク性能値の12倍あればよいことになります．

実際にはキャッシュ・メモリ内の実効性能はメモリ・バ

ンド幅にかかわらず高く，科学技術計算のようにキャッシュ・メモリに演算対象のデータが入りきらないアプリケーションでは，キャッシュ・リフィルはかなり高い税金（オーバヘッド）になっています．各プロセッサのピーク性能の12倍とFSBバンド幅（OpteronとCellについてはメモリ・バンド幅）の比も**表2**に示します．

さて，一時期，世界最速を誇っていた地球シミュレータのプロセッサはどうなっているのでしょう．計算ノードと呼ばれる1グループに8個のベクトル・プロセッサと32個のメモリ・ユニットが合計256Gバイト/sで接続されています．1プロセッサでは，8GFLOPSのピーク性能と32Gバイト/sのFSBバンド幅になります．ピーク性能の12倍とFSBバンド幅の比は33.33%です．さらにメモリ・ユニットはフルパイプラインになっており，リード/ライトが滞りなく実行されます．

数字だけで比較すると，パソコンやゲーム機で使われているマイクロプロセッサは地球シミュレータの足元にも及びませんが，コストとのトレードオフを考えると，パソコンやゲーム機で同じ構成をとるわけにもいきません．こうしたマイクロプロセッサでもある程度の実効性能が確保できるのは，キャッシュ構造のくふうやソフトウェアの最適化が図られているためだと思われます．

さて，このようにプロセッサの性能改善の方法はひと通りではありません．以下では医療機器における市場要求を説明した後，筆者らが開発した画像処理プロセッサ（XTrillion3.0）の概要を紹介します．本プロセッサは，前ページで述べた**1）～3）**の方法を組み合わせて性能向上を実現しています．

● ハードで実現するか，ソフトで実現するか

数値演算を必要とする装置を開発するとき，その処理をハード・ワイヤード論理で実現するか，ソフトウェア（とプロセッサ）で実現するかは，とても重要な検討項目です．筆者らはおもに医療関係の画像処理，および画像処理前の正規化を行う数値演算を得意としています．例えば，医療・工業用CT（コンピュータ断層撮影）などで必要になる正規化処理や画像処理をハード・ワイヤード論理で実現したり，ソフトウェアで実現したりしています．そのためのカスタムLSIを，筆者らは開発しています．

ここで，なぜ，こうしたLSIをわざわざ開発する必要があるのかを説明します．現在主流となっているGPU（例えば，米国NVIDIA社やカナダATI Technologies社の3次元グラフィックスLSI）の多くは，サーフェイス・レンダリングと呼ばれるモデリング手法を採用しています．これは，面の集合に基づいて視点や光源，反射を演算し，画像を生成する手法です．一方，医療機器では，サーフェイス・レンダリングではなくボリューム・レンダリングが使われます．処理の単位は面ではなくキューブ（立方体）です．そのため，市販のGPUをそのまま採用しにくいという事情があります．

次に，医療機器で必要となる数値演算の概要について説明します（**図1**）．

CTによって取得するデータは，レントゲン写真のようなものです．これをぐるりと360度回転させながら，約1,000枚程度の連続撮影を行います．一つのCT画像を取得するために，30秒～1分程度の撮影時間がかかります．このレントゲン写真を使って演算し，キューブ配列の空間に変換することを「再構成演算（reconstruction）」と呼びます．再構成演算はとても負荷の重い処理です．データ量にもよりますが，処理に10分～20分程度の時間がかかります．

CTなどでキャプチャした人体データの再構成演算が終わったら，その結果に対してレンダリング（描画）を行います．この処理は「可視化（visualization）」と呼ばれています．医療の現場では，心臓や脳，血管，消化器官などの内臓を

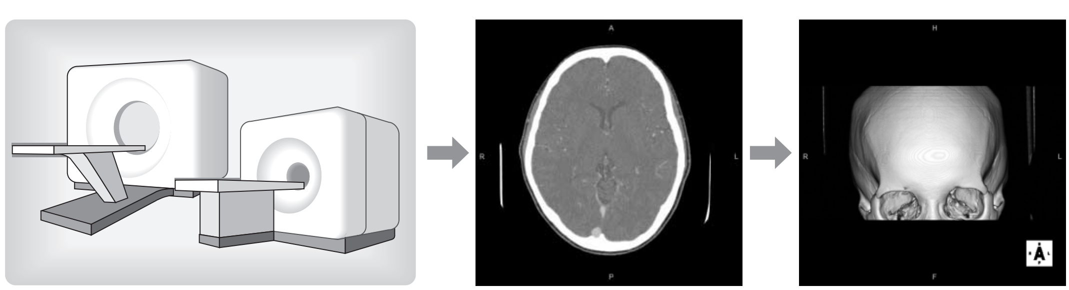


図1
医療機器における画像処理の流れ
CTやMRI（磁気共鳴診断装置）によって画像を取り込む．これに対して，再構成演算，可視化といった処理を順番に施す．

観察します．皮膚や骨などは透過処理を行いながらレンダリングします．その処理の後，問題箇所の観察が行えるようになります．このときのレンダリング処理では，例えば512×512×512＝134,217,728ボクセル（ボクセルは立方体の集合数）の空間を演算します．こうしたレンダリングを高速に実行するため，例えば筆者らはハードウェア・ボリューム・レンダリング・ボード「VolumePro 1000」を開発・出荷しています．

● 目的を明確にして，ハード，ソフトの両方を開発

医療機器に使われるボリューム・レンダリングは，バランスが重要です．画質ばかり追求すると，とてつもない演算時間が必要になります．速度を上げるためには，画質を落とさなければなりません．医療機器では高画質と高性能の両立が求められます．

そこで，筆者らはこうしたレンダリング処理をハード・ワイヤード論理で実現しています．例えば，上述のVolume Pro 1000は，パソコン・ベースのシステムと比較したとき，10倍程度の性能を備えています．画質をさほど落とさずに，透過処理やクリップ処理を行いながら30フレーム/sのレンダリングを行えます．

ただし，ハード・ワイヤード論理による設計では，多種多様な要求には対応できません（機能を自由にカスタマイズできない）．そこで同時並行で開発しているのが，ソフトウェアによって処理を実現するXTrillionプロセッサです．これは，再構成演算をはじめとする医療機器の各種数値演算に対応するLSIで，ソフトウェア処理における最高性能をねらって開発しています．

ハード・ワイヤード論理による3次元画像処理ボードとソフトウェア処理によるプロセッサには，それぞれ明確な目的があり，両方の開発を続けてきました．XTrillionプロセッサについては，バージョン1.0，1.5と開発され，2001年にバージョン2.0がリリースされて，現在，医療機器メーカに提供されています．また，最新版のXTrillion3.0が，現在，リリース直前の状態にまで来ています．

● 8個のベクトル・プロセッサを相互に全直結

2001年のXTrillion2.0のリリースからほどなく，筆者らは次期プロセッサの開発に着手する必要に迫られました．これはプロセッサの宿命でもありますが，継続的に演算能力を向上することが求められるためです．

最初にアーキテクチャの検討から始めました．当時，もっともポピュラな半導体プロセスは0.18 μmであり，半導体プロセスに関する知識や経験の乏しい筆者らが，Intel社のように1GHzで動作するプロセッサを開発することは「夢のまた夢」でした．当時，筆者らは，「似たようなRISCプロセッサを開発しても意味がない」と考えていました．この発想が今回のプロセッサ開発の根幹となり，自然に多並列化を目ざすことになりました．

最初は，当時のPentiumプロセッサなどの性能を考慮して，166MHz動作で8GFLOPS程度の性能を目標としました．多並列化することで，多数のALUが必要になります．単純な割り算でも，ALUは（8GFLOPS÷166MHz＝）48個必要です．この48個をどうやって接続するか，ここがもっとも肝心なところでした．

48個のALUを単純に一つの命令で管理すると，48SIMDプロセッサになってしまいます．また，もともとRISCプロセッサにはしないと決めていましたが，実効性能も高くしなければなりません．実効性能が高いということは，「プロセッサが止まらない．いや，止まってはならない」と考えを巡らせました．つまるところ，すべての演算がスループット1であれば問題ないはずです．そこで，パイプライン化，すなわちベクトル型を採用することにしました．

ベクトル・プロセッサの歴史は長く，初めて商業的に成功したベクトル・プロセッサ搭載マシンは，1976年にSeymour Cray氏が開発した「CRAY-1」です．ベクトル・プロセッサの設計経験のない筆者らが「スーパ・コンピュータの父」と呼ばれたCray氏を越えられるわけがありません．ただし，ベクトル・プロセッサの問題の多くは，複雑なスケジューラ回路に集中しています．幸運なことに，筆者らは優秀なソフトウェア・エンジニアに恵まれており，スケジューリングをコンパイラ任せにすることで，この問題を回避することにしました．

しかし，48個のALUを接続するという問題はまだ残っています．ここで，設計可能な方法を見つけ出すことに多くの時間を費やしました．結果的には，これらを8分割して，1CPUブロックにつき6個のALUを搭載することにしました．並列プロセッサなので，当然のことながら6個のALUを一つの命令で管理します．そして今度は，8個のCPUブロックどうしの通信方式が問題となります．

ここでは思い切りよく全直結とすることにしました（**図2**）．この思い切りがなかったら，おそらく目標性能は達成でき

なかったと思います．

● RISCを集積，並列ベクトル型MIMDプロセッサに

次に，本CPUブロックの仕様について説明します．

かりに，ベクトル演算用ALU（以下，VALU）をすべて同じ処理を行える仕様にしたとすると，天文学的な規模のゲートが必要になります．ベクトル・プロセッサの特徴ともいえるチェイニングができないと，性能もがた落ちです．SIMDのように同じ演算を6個のALUが繰り返すようでは，性能向上の効果が下がります．そこで，以下のような構成に決めました．

- 乗算加減算VALU×4個
- 除算加減算VALU×1個
- 超越関数演算VALU×1個（後に，VFUNと改名）
- 8Kワード・ベクトル・レジスタ×1個
- 命令RAM×1個
- チェイニング・テーブル×64個（当初は16個だったが，テープアウト間近に64個に変更した）
- コマンド・プロセッサ（簡単なCPU）×1個

これらを内蔵するCPUブロックを，筆者らは「VPIPE（ブイ・パイプ）」と呼んでいます．それぞれの演算器は別々に演算を実行できる命令仕様としましたが，結局，これらをまとめるスケジューラが必要となり，当初の意気込みとは少し外れて，RISCプロセッサを1個搭載することにしました．つまり，合計で9個のCPU（ブロック）を内蔵する並列演算チップとなり，この時点で並列ベクトル型MIMDプロセッサとなりました．

● フィジカル合成ツールやレイアウト設計サービスを利用

ここで，少し苦労話を紹介します．仕様が決まり，本格的な設計を開始しましたが，開発人員が明らかに不足していました．ASIC開発の経験上，もっとも人員が必要なのは検証と配置配線です．

このとき筆者らの目に留まったのが，米国Synplicity社のフィジカル合成ツール「Amplify ASIC」でした．一般的な論理合成ツールとは異なり，物理的な制約を考慮しつつ，仮の配置配線まで実行してくれます．また，チップになったときと比べてタイミングの誤差が小さく，合成結果が目標に達していれば配置配線の手戻りの回数が最小で済む，との触れ込みでした．筆者らは，すぐにこのツールに飛びつきました．ほどなく試作のベクトルALUの設計がしあがり，Synplicity社の協力を得て合成を行ったところ，166 MHzを超える結果が出ました．このころ，3.2GHz動作のPentium 4プロセッサがすでに市場に出回り，もっと高性能にする必要に迫られたこともあり，目標クロックをそれまでの倍の333MHzに変更することになりました．当時，フィジカル合成の結果に自信満々で楽勝ムードでしたが，この高い目標に後々最後まで苦しめられることになりました．

次に，この巨大なプロセッサの検証をどうやって行うか，これも悩みに悩みました．最終的には，プロセッサ全体が入るFPGA基板を作ってしまいました（**写真1**）．FPGAの総数は74個で，1,000Wの電源で動作します．自称（!?）世界最大のPCIボードができ上がりました．このような検証用基板は2度と作りたいとは思いません．基板のアートワークと製造を担当した協力会社の尽力により，なんとか完成させることができました．パソコン上の論理シミュレーションで333Hzで動作していたものが，33.3MHz（実機の

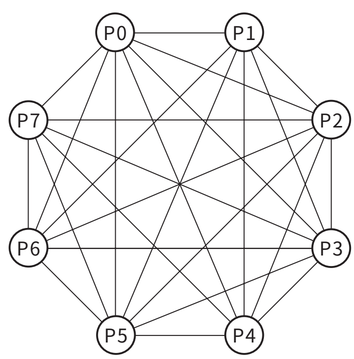


図2 プロセッサ間の結合のイメージ

8個のプロセッサどうしを，思い切りよく全直結した．

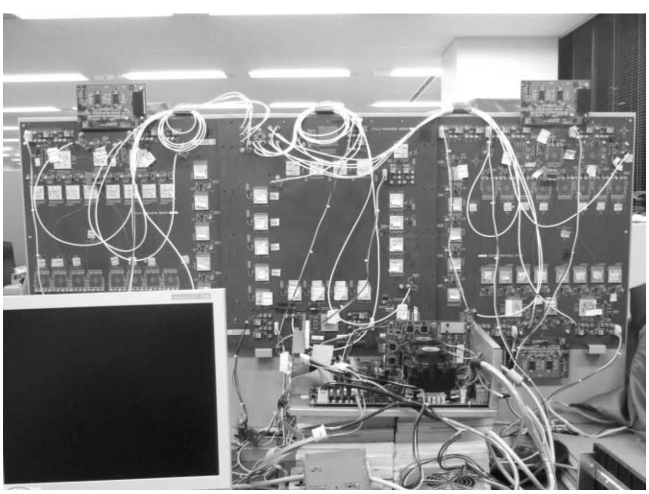


写真1　プロセッサ全体が入るFPGA基板を開発

1/10の動作周波数．最高で50MHzの動作も確認した）で動くようになり，検証速度はシミュレーションの10万倍に改善しました．

次の問題は，LSIにするためには，配置配線を行わなければならないことでした．ここで登場したのが，台湾のAlchip Technologies社でした．彼らは整備された設計方法論とマン・パワーを持ち合わせた企業でした．初めての打ち合わせを行ったときに非凡さを感じ，「ここしかない」と上司を説得しました．

とは言え，一般に利用できるCMOSプロセスを使って333MHz動作を実現することはかなり困難でした．333 MHzで確実に動作させるためには，さまざまな要因を検討し，最悪ケースでも363MHzで動作するネットリストを用意する必要がありました．この目標を達成するために，また血のにじむような苦労を重ねた結果，ネットリストをAlchip社に提出した後は，タイミング違反に起因する配置配線の手戻りはいっさい発生しませんでした．Amplify ASICと巨大なFPGA基板，Synplicity社の深夜をいとわないサポート，Alchip社の豊富な設計経験，開発メンバの不眠不休の努力と不退転の覚悟がそろい，ES（engineering sample）チップが2005年11月に到着しました（下掲のコラム「プロセッサ開発の苦労」を参照）．

チップは一発で動作し，感動はひとしおでした．もちろんチップは目標どおり333MHzで動作し，Amplify ASICの高精度とAlchip社の配置配線の優秀さも確認できました．

● 実効性能がRISCより高いことを確認

本プロセッサの概要を**図3**に示します．基本動作の検証は，ESチップの到着から6時間程度で済みました．これも設計段階で，検証につぐ検証を実施した苦労の結果であると考えています．

いちばん問題となる実効性能ですが，**表3**のようになっています．この数値は単精度浮動小数点演算の結果です．



（a）チップの外観

項　目	仕　様
電源電圧	コア部：1.2V，I/O部：3.3V
消費電力	最大18.9W
パッケージ	1155ピンFBGA，フリップチップ実装
VPIPE部内蔵メモリ	VREG：256Kバイト（32Kバイト×8） VMEM：64Kバイト（16Kバイト×4） 命令RAM：512Kバイト（64Kバイト×8） チェイニング・テーブル：64
RISC部内蔵メモリ	キャッシュ：16Kバイト ベクトル・ブリッジ・メモリ： 128Kバイト（16Kバイト×8）
SDRAMインターフェース	64ビット×2，最大512MバイトのSDRAMと接続
SRAMインターフェース	64ビット×4，最大128MバイトのSRAMと接続
動作周波数	コア部：最大333MHz 外部メモリ：最大166MHz （内部クロックと同じ，または1/2の周波数）
外部I/Oバス	64ビット×2ポート，コア部と非同期動作
理論ピーク性能 （VALU部＋VFUN部）	13320MFLOPS＋2664MFLOPS, 53280MOPS＋2664MOPS

MFLOPS：million floating-point operations per second
MOPS：million operation per second

（b）おもな仕様

図3
XTrillion3.0の概要
単精度浮動小数点演算で16GFLOPSのピーク性能を達成した．

コラム　プロセッサ開発の苦労

今回のプロセッサの開発は，8名（このうち，ソフトウェア担当が2名）で行いました．チップの製造やFPGA基板の製作については外部の企業に委託しましたが，仕様検討やコーディング，シミュレーション，検証，フィジカル合成（論理合成）などはすべて社内で行いました．工程数は最終的に600項目に及び，ほとんど会社に泊り込んでいた時期もありました．

プロセッサ開発は，ハードウェアのみならず，検証の数学的解決やコンパイラ開発，工程管理など，困難を極めます．ここで，もっとも大事なことは「作りたい」と思う気持ちです．「作りたい」という強い意志さえあれば，けっこう何とかなるものです．

振り返ってみると，開発期間中に結婚を果たしたメンバは結婚式の前々日の朝までコーディングとシミュレーションを繰り返し，そのまま故郷行きの飛行機に乗りました．会社にりっぱなお風呂セットを用意したメンバ，寝袋に入る気力もなくそのまま抱きかかえて寝てしまったメンバ，作業に夢中になりすぎて1日食事を取るのを忘れたメンバ，そして終電に乗り遅れて会社に戻ってきたメンバ…．申しわけないと思いつつ久しぶりに自宅に帰ると，自分の記憶ではオムツ付きでヨチヨチ歩きだったわが子が，普通に立って歩き回っていました．このときは，さすがにわれに返りました．

よくこんなことができたものです．メンバ，および家族のみなさんの理解と協力（あきらめ？）に感謝するしだいです．このプロセッサは，コンピュータを愛する，少しアブナイ人々によって作られました．

表3　ピーク性能と実効性能

プロセッサ	動作周波数	ピーク性能	実効性能	実効性能/ピーク性能
Xeon	2.8GHz	11.2GFLOPS	6.5GFLOPS 注	58%
Opteron 246	2.0GHz	8.0GFLOPS	4.6GFLOPS 注	58%
PowerPC G5	2.0GHz	16.0GFLOPS	9.1GFLOPS 注	57%
XTrillion3.0	333MHz	15.9GFLOPS	10.2GFLOPS	64%

注：http://www.fftw.org/を参照.

比較対象のプロセッサがやや古いのですが，ここでは$N=1024$，Radix-2，1D Complex Single Precision Floating Point FFT（fast Fourier transform）をベンチマークとして利用しました.

アーキテクチャが異なりますが，RISCプロセッサの実効性能とピーク性能の比は57%～58%，本プロセッサでは64%となっています．FFTに使用しない超越関数演算器（VFUN）を入れなければ，ピーク性能は13.33GFLOPS，実効性能は77%となります.

実際のアプリケーションでは，これほどRISCプロセッサにとって好条件になる演算はそう多くありません．一方，本プロセッサは，ベクトル化できる演算であれば高効率で実行します.

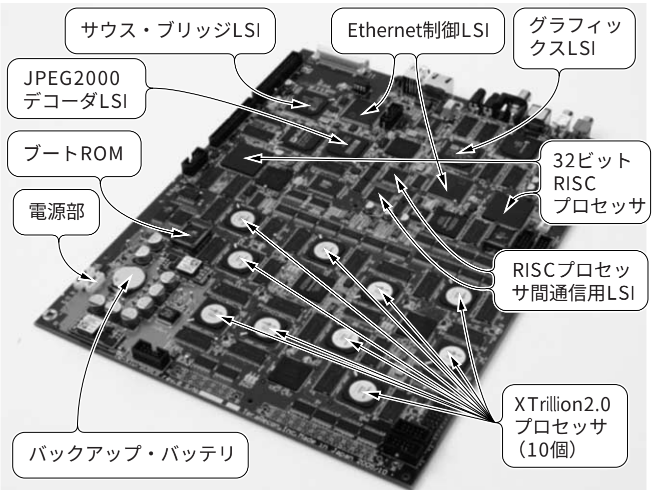
● x86プロセッサと役割を分担

x86アーキテクチャにおいても，組み込み機器への搭載を視野に入れた消費電力10W以下のプロセッサが登場しています．組み込みシステムにも並列プロセッサによる並列処理の波が押し寄せてきています．筆者らは，今回開発したプロセッサがx86プロセッサにとって代わるとは考えていません．RISC CPUを内蔵しているのでLinux OSなども実行可能ですが，スカラ演算となるとx86の足元にも及びません．また，巨大なメモリ空間の管理やセキュリティ・ビットなど，OSを安全に動作させるためのしくみもぜい弱です．本プロセッサはあくまでも，ベクトル化が可能な演算で真価を発揮するプロセッサなのです.

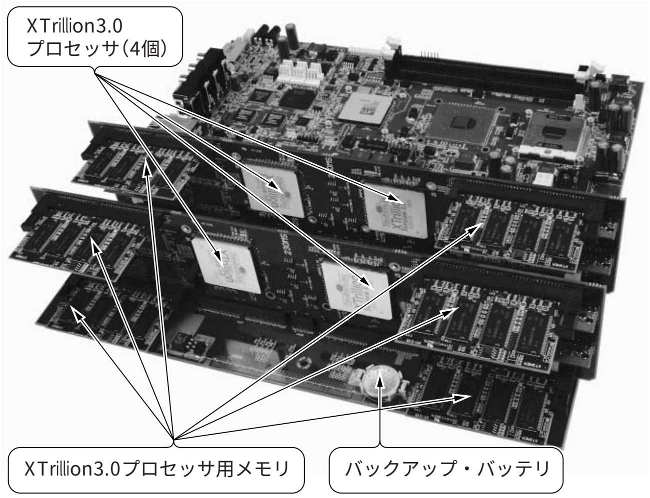
本プロセッサは，既存のシステムに追加で組み込む用途をねらって開発しています．そのしくみについて，ポイントを説明します.

実際の信号処理や画像処理などでは前処理があり，A-Dコンバータなどでディジタル化されたデータを次段へ引き渡します．本プロセッサではこうした処理の流れを前提としており，X-BUSと呼ばれるバス・インターフェースを二つ用意しています．ここで，X-BUSのバス・クロックはコア部のクロックと同期している必要はありません．本プロセッサの内部で非同期乗り換えを行っているので，接続はスムーズに行えます．処理後の結果については，PCIバスなどのDMA転送で外部のホスト・マシン（パソコン）に引き出すことができます.

各種の信号処理が終わったら可視化を行い，ユーザに対



（a）1020BE



（b）430PA

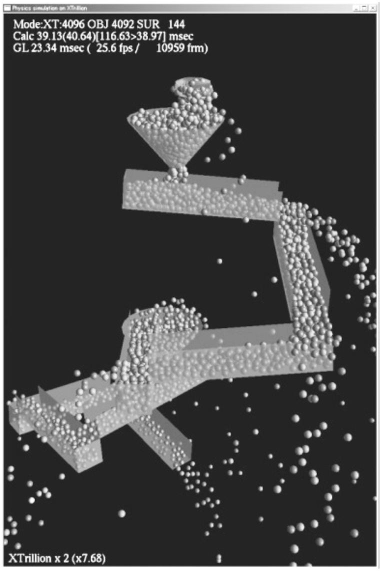
写真2　搭載ボードの例

（a）は，2個のRISCプロセッサと10個のXTrillion2.0を1枚の基板に搭載している．基板サイズは320mm×250mm．もともと並列化を想定して設計されているプロセッサなので，こういった構成は容易に実現できる．（b）は，1個のPentium Mプロセッサと4個のXTrillion3.0を搭載している．ボード全体の処理性能は64GFLOPS.

図4
処理性能の評価
誌面ではわかりにくいが，演算を担当しているプロセッサ別に色分けして表示させている．

ボール数	Xeonのみ	XTrillion3.0を1個追加	XTrillion3.0を2個追加
64個	0.52ms	3.64ms（0.14倍）	7.35ms（0.07倍）
128個	1.15ms	3.64ms（0.30倍）	7.35ms（0.16倍）
256個	2.82ms	3.64ms（0.74倍）	7.35ms（0.38倍）
512個	7.92ms	3.64ms（2.18倍）	7.35ms（1.08倍）
1,024個	24.55ms	7.51ms（3.27倍）	7.35ms（3.34倍）
2,048個	83.85ms	23.77ms（3.52倍）	14.69ms（5.71倍）
4,096個	303.44ms	—	39.02ms（7.78倍）

（**a**）ボール数と演算時間



（**b**）シミュレーション

して結果を出力します．OSの管理などを含めて，ユーザ・インターフェース処理はホスト・マシンが担当することを想定しています．また，ホスト側が信号処理などにリソースを割く必要はありません．つまり，ホスト・マシンのプロセッサに，それほど高性能なものを使う必要がないのです．

一般に，万能のプロセッサは存在しません．システム設計では，プロセッサの特徴を考慮して選択し，組み合わせる必要があります．汎用のマイクロプロセッサだけでは不足する演算能力を補うため，本プロセッサのような数値演算用プロセッサを多並列化して，所望の性能を達成するのも一つの方法だと思います．この考えかたに基づいて，筆者らはいくつかの並列プロセッサ・システム（ボード）を設計してきました．その例を**写真2**に示します．

● x86だけの場合と比較して，処理性能は最大7.78倍に

最後に本プロセッサの性能評価の例を示します．ここではPCIボードに2個の本プロセッサを実装した基板（230AX）を使いました．

図4は，3GHzのクロックで動作するIntel社のXeonプロセッサだけで処理した場合と，Xeonプロセッサと1個または2個の本プロセッサを組み合わせて処理した場合の比較です．落下してくるボールは最大4,096個で，壁との衝突と重力を計算し続けています．表示にはOpenGLライブラリを使用しています．この部分だけはXeonプロセッサの力を借りなければなりません．それ以外の計算はすべて本プロセッサで処理しています．

本プロセッサを2個使った場合，最大7.78倍の処理性能の向上が見られることがわかります．ボールが少ない場合はXeonのほうが高速ですが，玉が増えるにしたがって本プロセッサを追加したほうが有利になります．これは，キャッシュ・リフィルが頻発し，実効性能がどんどん低下していることが原因であると考えられます．ボールの数や壁の数が増えれば増えるほど，この差は広がっていきます．これがベクトル・プロセッサの特徴です．

Y
米国TeraRecon社

<筆者プロフィール>
Y（ペン・ネーム）．台湾系コンピュータ・メーカ，画像処理関連ベンチャ，並列コンピュータ関連ベンチャを渡り歩く．コンピュータばかに拍車がかかり，医療機器メーカであるTeraRecon社がプロセッサ設計をやっていることだけを理由に，2000年に入社．自分でやらないと気が済まない自己中心型．地球シミュレータを分解してみたいとチャンスをねらっている．なお，TeraRecon社は業容拡大中につき，現在，エンジニアを随時募集中．詳しくは，http://www.terarecon.co.jp/を参照してください．問い合わせはinfo@terarecon.co.jpまで．